

DERWENT-ACC-NO: 1995-102787  
DERWENT-WEEK: 199514  
COPYRIGHT 1999 DERWENT INFORMATION LTD

TITLE: Software renewal method for communications network - updating and classifying software independent of physical memory configuration from remote location

PATENT-ASSIGNEE: NIPPON TELEGRAPH & TELEPHONE CORP[NITE]

PRIORITY-DATA: 1993JP-0191842 (July 7, 1993)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
JP 07028724 A	January 31, 1995	N/A	008	G06F 013/00

APPLICATION-DATA:

PUB-NO	APPL-DESCRIPTOR	APPL-NO	APPL-DATE
JP 07028724A	N/A	1993JP-0191842	July 7, 1993

INT-CL (IPC): G06F009/06; G06F013/00 ; H04M003/00 ; H04Q003/545

ABSTRACTED-PUB-NO: JP 07028724A

BASIC-ABSTRACT: The software renewal method involves using an operating system to remotely renew software through a communication network. An operation system controls the operation which carries out updating of the network elements. Each network element notifies the updating situation of the software renewal by state change to an operation system. The attribute expresses the cut return state on the gamma state from a low-speed memory media and cut return impossible state and a file memory which can be cut returned and a hard disk.

USE/ADVANTAGE - File renewal, location data change of software of network elements in communication network from remote location. Controls operation interface and control sequence becomes logical. Facilitates inter-operability between different hardware and software.

CHOSEN-DRAWING: Dwg.1/18

TITLE-TERMS:

SOFTWARE RENEW METHOD COMMUNICATE NETWORK UPDATE CLASSIFY  
SOFTWARE INDEPENDENT  
PHYSICAL MEMORY CONFIGURATION REMOTE LOCATE

DERWENT-CLASS: T01 W01

EPI-CODES: T01-F05; T01-H07C; W01-B02A1; W01-C02A7;

SECONDARY-ACC-NO:

Non-CPI Secondary Accession Numbers: N1995-080898

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-28724

(43)公開日 平成7年(1995)1月31日

(51)Int.Cl.<sup>\*</sup>  
G 0 6 F 13/00 3 5 1 H 7368-5B  
9/06 5 4 0 A 9367-5B  
H 0 4 M 3/00 E 8426-5K  
H 0 4 Q 3/545 8843-5K

F I

技術表示箇所

審査請求 未請求 請求項の数2 FD (全8頁)

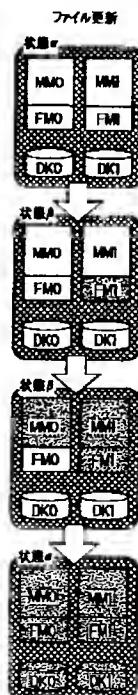
(21)出願番号	特願平5-191842	(71)出願人	000004226 日本電信電話株式会社 東京都千代田区内幸町一丁目1番6号
(22)出願日	平成5年(1993)7月7日	(72)発明者	高橋 和秀 東京都千代田区内幸町一丁目1番6号日本 電信電話株式会社内

(54)【発明の名称】 ソフトウェア更新制御方法

(57)【要約】

【目的】 本発明はO p SとNEの独立な発展を可能として相互接続性を保証し、NE側の更新種別の変更あるいは追加、新機種のNEの導入に対して柔軟に対応するソフトウェア更新制御方法を提供することを目的とする。

【構成】 本発明は上記目的を達成するために、ソフトウェア更新の更新状況を、オペレーションシステムのプログラムのクラスオブジェクト内の更新状態を表す属性の値により表現する。また切戻し不可能状態である状態 $\alpha$ 、ファイルメモリ等の高速記憶媒体から切戻し可能状態である状態 $\beta$ 及びハードディスク等の低速記憶媒体から切戻し可能状態である状態 $\gamma$ で表現する。よって、NEに対するO p Sの制御操作インターフェースをNEの機種、更新種別及びベンダに非依存となるために、制御操作インターフェース及び制御操作シーケンスの論理化により、O p SとNEの独立な発展を可能とし、相互接続性を保証できる。



## 【特許請求の範囲】

【請求項1】 オペレーションシステムが交換機等のネットワークエレメントを保守するために、オペレーションシステムが遠隔から通信網を介してネットワークエレメントのソフトウェア更新を実施するためにネットワークエレメントを制御するソフトウェア更新制御方法において、

ソフトウェア更新の更新状況を、オペレーションシステムのプログラムのクラスオブジェクト内の更新状態を表す属性の値により表現し、かつ切戻し不可能状態である

状態 $\alpha$ 、ファイルメモリ等の高速記憶媒体から切戻し可能状態である状態 $\beta$ 及びハードディスク等の低速記憶媒体から切戻し可能状態である状態 $\gamma$ で表現し、

オペレーションシステムがネットワークエレメントに対して行う制御操作を実行 $\beta$ 、実行 $\gamma$ 、バックアップ $\alpha$ 、

バックアップ $\gamma$ 及び切戻しにより表現し、

オペレーションシステムはソフトウェア更新の起動契機によりクラスオブジェクトのインスタンスを生成し、

ネットワークエレメントがオペレーションシステムに対してソフトウェア更新の更新状況を状態変更により通知し、

オペレーションシステムはネットワークエレメントからの通知を契機にクラスオブジェクトのインスタンスの属性値を変更することにより、ソフトウェア更新の更新状況を管理し、インスタンスの状態遷移によりソフトウェア更新を制御することを特徴とするソフトウェア更新制御方法。

【請求項2】 他のネットワークエレメントの導入や他のソフトウェア更新種別の追加に対しては前記クラスオブジェクトのサブクラス化により状態を表す属性を追加する請求項1記載のソフトウェア更新制御方法。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明はソフトウェア更新制御方法に関し、特に交換機等のネットワークエレメント（以下NEと略す）の保守運用を行うシステムであるオペレーションシステム（以下OoSと略す）がNEを保守する環境においてOoSが遠隔から通信網を介してNEのソフトウェア更新（ファイル更新、パッチ投入、所データ変更等）を実施する場合、OoSがNEを制御する方法に関する。

## 【0002】

【従来の技術】現在、図13及び図14に示すような様々な物理メモリ構成を持ったNEが存在する。図13はメインメモリ（以下MMと略す）、ファイルメモリ（以下FMと略す）及びハードディスク（以下DKと略す）の二重化メモリ構成を持つNEの例を示す図である。また図14はMM及びDKの二重化メモリ構成を持つNEの例を示す図である。

【0003】OoSがNEに対して行うソフトウェア更

新の業務には、交換機のソフトウェアバグの修正や機能追加を目的としてシステムファイル全体を入れ替えるファイル更新と、交換機のソフトウェアバグの修正や機能追加を目的としてシステムファイルを部分的に修正するために書き替えるパッチ投入と、交換機の装置増設、回線増設、局番変更等に伴って所データを部分的に変更する所データ変更等の業務がある。これらの業務シーケンスの中で、OoSはNEに対して以下のような様々な制御操作を行う。

【0004】図15は図13の物理メモリ構成を持つNEのソフトウェア更新シーケンスのうちファイル更新を示し、図16はパッチ投入を示し、図17は所データ変更を示す。また図18は図14の物理メモリ構成を持つNEのソフトウェア更新シーケンスのうちファイル更新を示す。

【0005】まず図15に示す図13の物理メモリ構成を持つNEのファイル更新シーケンスについて説明すると、最初のFUP NEW/の制御操作により1系のFM1に新ファイルがロードされる（この処理は図中編掛で示す）。次のCNV FUP OK/の制御操作によりファイル更新再開処理が行われ、両系のMM0とMM1に新ファイルがロードされ、NEは1系で立ち上がる。最後のFUP FIN/の制御操作により0系のFM0に新ファイルがコピーされ、引き続いて両系のDK0とDK1にバックアップファイルが収集される。

【0006】次に図16に示す図13の物理メモリ構成を持つNEのパッチ投入シーケンスについて説明すると、最初のMLD PAT,PUP/の制御操作により新パッチが両系のMM0とMM1に投入される。次のMLD PAT,SUP/の制御操作により新パッチが両系のFM0とFM1に新ファイルに投入される。最後のFST BFM,LAF/により両系のDK0とDK1にバックアップファイルが収集される。

【0007】図17に示す図13の物理メモリ構成を持つNEの所データ変更シーケンスについて説明すると、最初のTCA/の制御操作により両系のMM0とMM1の中間テーブル（図中太い罫線で示す）の所データが変更される。次のTCH NEW/の制御操作により両系のMM0とMM1の中間テーブルが変更される。ODB/の制御操作により両系のFM0とFM1が変更され、最後のFST BFM,LA F/により両系のDK0とDK1にバックアップファイルが収集される。

【0008】図18に示す図14の物理メモリ構成を持つNEのファイル更新シーケンスについて説明すると、最初のFUP NEW/の制御操作により1系のMM1に新ファイルがロードされる。次のCNV FUP OK/の制御操作によりファイル更新再開処理が行われ、NEは1系で立ち上がる。最後のFUP FIN/の制御操作により両系のDK0とDK1にバックアップファイルが収集される。

【0009】

【発明が解決しようとする課題】通常、従来のソフトウェア更新の制御方法では、更新種別に応じて、O p SがNEに対して行う制御操作インタフェース及び制御シーケンスは異なるため、従来例によれば、NEの仕様変更に伴い更新種別が変更あるいは追加される場合、O p Sの制御シーケンスを変更あるいは追加しなければならない。従って、O p S側のプログラムはNE側のプログラムの変更の影響を受けるため、NEとO p Sの独立な発展ができないという問題点がある。

【0010】またNEの物理メモリの構成に応じて、O p SがNEに対して行う制御操作インタフェース及び制御シーケンスが異なるため、O p Sが保守するNEの機種（例えば交換機の場合D70, ISM, D51, DMS10など）が増える場合、従来例によれば、O p Sの制御操作インタフェース及び制御操作シーケンスを追加しなければならない。従って、今後通信ネットワークの高度化にともない様々なベンダ（特に海外northern tel ecom, ericsson, ATTなど）がNEを提供するマルチベンダ環境を考えるとNEとO p Sの相互接続性が保証されないとする問題点がある。

【0011】本発明の目的は、制御操作インタフェース及び制御操作シーケンスの論理化により、O p SとNEの独立な発展を可能とし、相互接続性を保証することにある。従って、本発明におけるO p SはNE側の更新種別の変更あるいは追加、新機種のNEの導入に対して柔軟に対応することができる。

#### 【0012】

【課題を解決するための手段及び作用】本発明は前記問題点を解決するために、ソフトウェア更新の更新状況を、O p Sのプログラムのクラスオブジェクト内の更新状態を表す属性の値により表現し、かつ切戻し不可能状態である状態 $\alpha$ 、ファイルメモリ等の高速記憶媒体から切戻し可能状態である状態 $\beta$ 及びハードディスク等の低速記憶媒体から切戻し可能状態である状態 $\gamma$ で表現する。また、O p SがNEに対して行う制御操作を実行 $\beta$ 、実行 $\gamma$ 、バックアップ $\alpha$ 、バックアップ $\gamma$ 及び切戻しにより表現する。さらに、O p Sはソフトウェア更新の起動契機によりクラスオブジェクトのインスタンスを生成し、NEがO p Sに対してソフトウェア更新の更新状況を状態変更により通知する。そして、O p SはNEからの通知を契機にクラスオブジェクトのインスタンスの属性値を変更することにより、ソフトウェア更新の更新状況を管理し、インスタンスの状態遷移によりソフトウェア更新を制御する。

【0013】また、他のNEの導入や他のソフトウェア更新種別の追加に対してはクラスオブジェクトのサブクラス化により状態を表す属性を追加する。従って、本発

明は前記問題点を解決でき、NEに対するO p Sの制御操作インターフェースをNEの機種、更新種別及びベンダに非依存となるために、制御操作インターフェース及び制御操作シーケンスの論理化により、O p SとNEの独立な発展を可能とし、相互接続性を保証できる。

#### 【0014】

【実施例】以下、本発明の実施例について図面に基づいて説明する。はじめに、ソフトウェア更新の途中経過を、切戻し不可能状態（この状態を状態 $\alpha$ と称す）、ファイルメモリ等の高速記憶媒体から切戻し可能状態（この状態を状態 $\beta$ と称す）及びハードディスク等の低速記憶媒体から切戻し可能状態（この状態を状態 $\gamma$ と称す）の3状態により表現し、制御操作インターフェースを実行 $\beta$ 、実行 $\gamma$ 、バックアップ $\alpha$ 、バックアップ $\gamma$ 及び切戻しにより表現する。

【0015】図1は本実施例における、図13の物理メモリ構成を持つNEのソフトウェア更新シーケンスのうちファイル更新シーケンスを示し、図2は本実施例における所データ変更シーケンスを示し、図3は本実施例におけるバッチ投入シーケンスを示す。また図4は本実施例における、図14の物理メモリ構成を持つNEのソフトウェア更新シーケンスのうちファイル更新シーケンスを示す。なお、各シーケンスの説明は後述するものとする。

【0016】ここで、図5はソフトウェア更新クラスの継承関係を示す図であり、更新状態及び制御操作を拡張（詳細化）したい場合このクラスをサブクラス化してサブクラスで新しい更新状態及び制御操作を定義する。例えば「状態 $\beta$ 」を「状態 $\beta$ 1」と「状態 $\beta$ 2」に拡張（詳細化）したい場合、新しいクラスの「更新クラス1」を定義する。このクラスは「状態 $\beta$ 」を拡張した属性として「 $\beta$ 更新状態」を持ち、その値として「状態 $\beta$ 1」及び「状態 $\beta$ 2」をとる。図6～8にはクラスオブジェクトの継承関係を示す。図6～8及びその状態遷移規則を示す下記の表1に示すように、属性として更新状態（値： $\alpha$ 、 $\beta$ 、 $\gamma$ ）、交換機に対するO p Sの制御操作（インターフェース）として実行 $\beta$ 、実行 $\gamma$ 、バックアップ $\alpha$ 、バックアップ $\gamma$ 及び切戻しを持つソフトウェア更新クラスをスーパーカラスとして定義する。また、2つ

ある状態 $\beta$ を区別するために、属性として更新状態 $\beta$ （値： $\beta$ 1、 $\beta$ 2）、制御操作として実行 $\beta$ 2を持つ更新クラス1を、状態 $\gamma$ を区別するために、属性として更新状態 $\gamma$ （値： $\gamma$ 1、 $\gamma$ 2）、制御操作として実行 $\gamma$ 2を持つ更新クラス2をソフトウェア更新クラスのサブクラスとして定義する。

#### 【0017】

【表1】

操作前		操作	操作後	
現状態	ファイル種別		次状態	ファイル種別
状態 $\alpha$	旧ファイル	実行 $\beta$	状態 $\beta$	新ファイル
状態 $\beta$	新ファイル	バックアップ $\gamma$	状態 $\beta$	新ファイル
状態 $\alpha$	旧ファイル	実行 $\gamma$	状態 $\gamma$	新ファイル
状態 $\gamma$	新ファイル	バックアップ $\alpha$	状態 $\alpha$	新ファイル
状態 $\beta$	新ファイル	切り戻し	状態 $\alpha$	旧ファイル
状態 $\gamma$	新ファイル	切り戻し	状態 $\alpha$	旧ファイル

【0018】Op Sのプログラムに、これらのクラスオブジェクト及び状態遷移規則を配備し、NEのプログラムに状態遷移規則を配備する。

【0019】保守者等が、Op Sに対して、目的のソフトウェア更新を指示すると、Op Sは対応するインスタンスを生成する。初期状態は更新状態が $\alpha$ 状態であるときとする。Op Sが、NEに対して対応する制御操作を行うと、NEは対応するメモリの更新を行い、状態遷移規則から次に状態を決定して状態遷移する。次に、NEが状態遷移した旨をOp Sに対して通知すると、Op Sはインスタンスの更新状態を変更し、状態遷移規則から次の制御操作を決定して、NEに対して制御操作を行う。目的のソフトウェア更新が終了し、インスタンスの更新状態が再び $\alpha$ 状態となるまで、以下同様に繰り返す。

【0020】図9は、本実施例における、図13に示す物理メモリ構成を持つNEについてのファイル更新シーケンスを示す図である。同図は図1に対応するものであり、同図において、最初の実行 $\beta$ の制御操作により、1系のFM1に新ファイルがロードされ、インスタンスは状態 $\beta$  ( $\beta$ 1) に遷移する。次の実行 $\beta$ 2の制御操作により、ファイル更新再開処理が行われ、両系のMM0, MM1に新ファイルがロードされ、NEは1系で立ち上がり、インスタンスは状態 $\beta$ 2に遷移する。最後の実行 $\alpha$ の制御操作により、0系のFM0に新ファイルがコピーされ、引き続いて両系のDK0, DK1に前日ファイル及び保証ファイルが収集され、インスタンスは状態 $\alpha$ に遷移する。

【0021】図10は、本実施例における、図13に示す物理メモリ構成を持つNEについてのパッチ投入シーケンスを示す図である。同図は図3に対応するものであり、同図において、最初の実行 $\gamma$ の制御操作により、新パッチが両系のMM0, MM1に投入され、インスタンスは状態 $\gamma$  ( $\gamma$ 1) に遷移する。次の実行 $\gamma$ 2の制御操作により新パッチが両系のFM0, FM1に投入され、インスタンスは状態 $\gamma$ 2に遷移する。最後の実行 $\alpha$ により両系のDK0, DK1に前日ファイルが収集され、インスタンスは状態 $\alpha$ に遷移する。

【0022】図11は、本実施例における、図13に示す物理メモリ構成を持つNEについての所データ変更シーケンスを示す図である。同図は図2に対応するものであり、同図において、最初の実行 $\beta$ の制御操作により両系のMM0, MM1の中間テーブル（試験呼はこのテーブルを参照するが実呼は旧データを参照する）の所データが変更され、インスタンスは状態 $\beta$ に遷移する。次の実行 $\gamma$ の制御操作により両系のMMの本テーブルが変更され、インスタンスが状態 $\gamma$  ( $\gamma$ 1) に遷移する。実行 $\gamma$ 2の制御操作により両系のFM0, FM1が変更され、インスタンスは $\gamma$ 2に遷移する。最後の実行 $\alpha$ により両系のDK0, DK1に前日ファイルが収集されインスタンスは状態 $\alpha$ に遷移する。

\*す物理メモリ構成を持つNEについての所データ変更シーケンスを示す図である。同図は図2に対応するものであり、同図において、最初の実行 $\beta$ の制御操作により両系のMM0, MM1の中間テーブル（試験呼はこのテーブルを参照するが実呼は旧データを参照する）の所データが変更され、インスタンスは状態 $\beta$ に遷移する。次の実行 $\gamma$ の制御操作により両系のMMの本テーブルが変更され、インスタンスが状態 $\gamma$  ( $\gamma$ 1) に遷移する。実行 $\gamma$ 2の制御操作により両系のFM0, FM1が変更され、インスタンスは $\gamma$ 2に遷移する。最後の実行 $\alpha$ により両系のDK0, DK1に前日ファイルが収集されインスタンスは状態 $\alpha$ に遷移する。

【0023】図12は、本実施例における、図14に示す物理メモリ構成を持つNEについてのファイル更新シーケンスを示す図である。同図は図4に対応するものであり、同図において、最初の実行 $\beta$ の制御操作により、1系のMM1に新ファイルがロードされ、インスタンスは状態 $\beta$  ( $\beta$ 1) に遷移する。次の実行 $\beta$ 2の制御操作により、ファイル更新再開処理が行われ、NEは1系で立ち上がり、インスタンスは状態 $\beta$ 2に遷移する。最後の実行 $\alpha$ の制御操作により、両系のDK1, DK2に前日ファイル及び保証ファイルが収集され、インスタンスは状態 $\alpha$ に遷移する。

#### 【0024】

【発明の効果】以上説明したように、本発明によれば、物理的なメモリ構成や更新種別に依存しないソフトウェア更新を行うための制御操作インターフェース及び制御シーケンスを実現できる。従って、Op Sが物理メモリ構成や更新種別を意識することなく、ソフトウェア更新を実現することができる。即ち、制御操作インターフェース及び制御シーケンスの論理化により、Op SとNEの独立な発展が可能となり、相互接続性が保証される。また、制御操作インターフェース及び制御シーケンスが変更される場合でも、スーパークラスのサブクラス化により、更新種別の追加あるいは変更、新機種のNEの導入に対して、柔軟に対応できるので、NEプログラムの変更に影響されるOp Sプログラムの変更は局在化され、容易に機能追加することが可能となり、Op Sプログラムの流用性が向上する。

## 【図1の簡単な説明】

【図1】本実施例におけるファイル更新シーケンスを示す図である。

【図2】本実施例における所データ変更シーケンスを示す図である。

【図3】本実施例におけるパッチ投入シーケンスを示す図である。

【図4】本実施例における、別の物理メモリ構成を持つNEのファイル更新シーケンスを示す図である。

【図5】ソフトウェア更新クラスの継承関係を示す図である。 10

【図6】クラスオブジェクトの継承関係を示す図である。

【図7】クラスオブジェクトの継承関係を示す図である。

【図8】クラスオブジェクトの継承関係を示す図である。

【図9】本実施例におけるファイル更新シーケンスを示す図である。

【図10】本実施例におけるパッチ投入シーケンスを示す図である。 20

す図である。

【図11】本実施例における所データ変更シーケンスを示す図である。

【図12】本実施例における、別の物理メモリ構成を持つNEのファイル更新シーケンスを示す図である。

【図13】MM、FM及びDKの二重化メモリ構成を持つNEの例を示す図である。

【図14】MM及びDKの二重化メモリ構成を持つNEの例を示す図である

【図15】従来のファイル更新シーケンスを示す図である。

【図16】従来のパッチ投入シーケンスを示す図である。

【図17】従来の所データ変更を示す図である。

【図18】従来の別の物理メモリ構成を持つNEのファイル更新シーケンスを示す図である。

## 【符号の説明】

MM メインメモリ

FM ファイルメモリ

DK ハードディスク

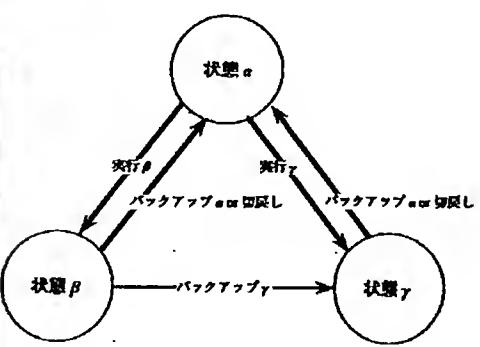
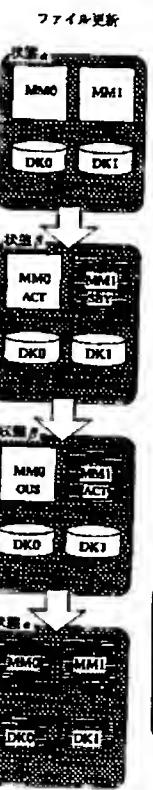
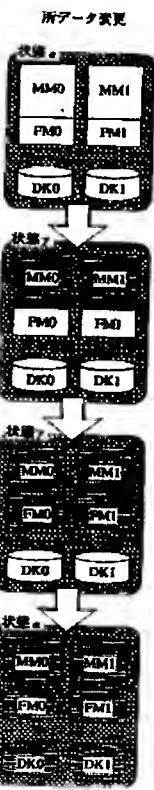
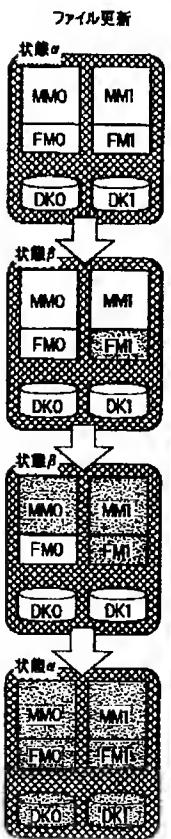
【図1】

【図2】

【図3】

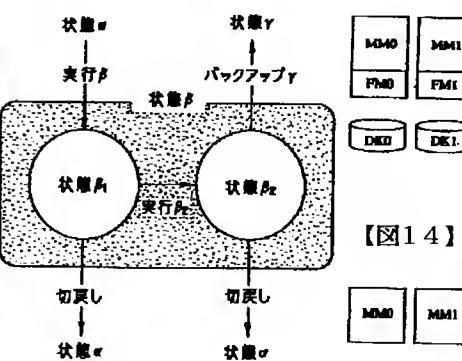
【図4】

【図6】

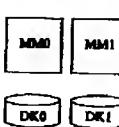


【図7】

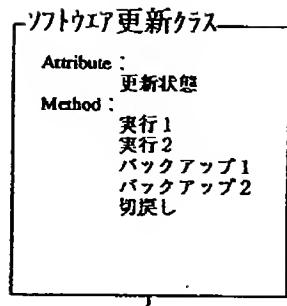
【図13】



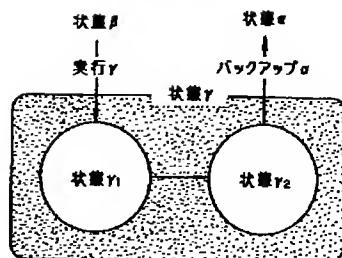
【図14】



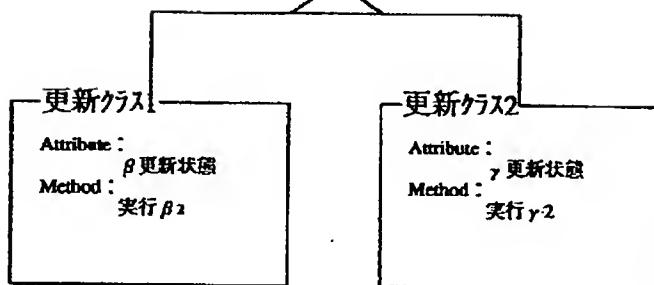
〔図5〕



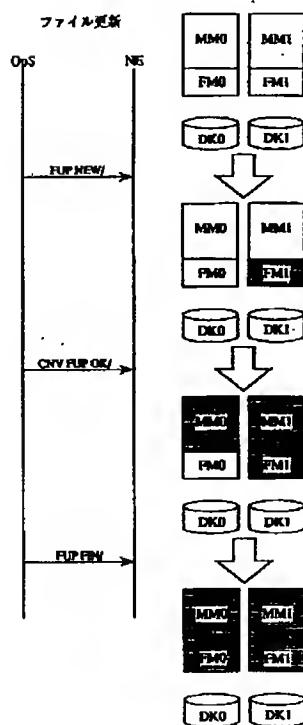
[图8]



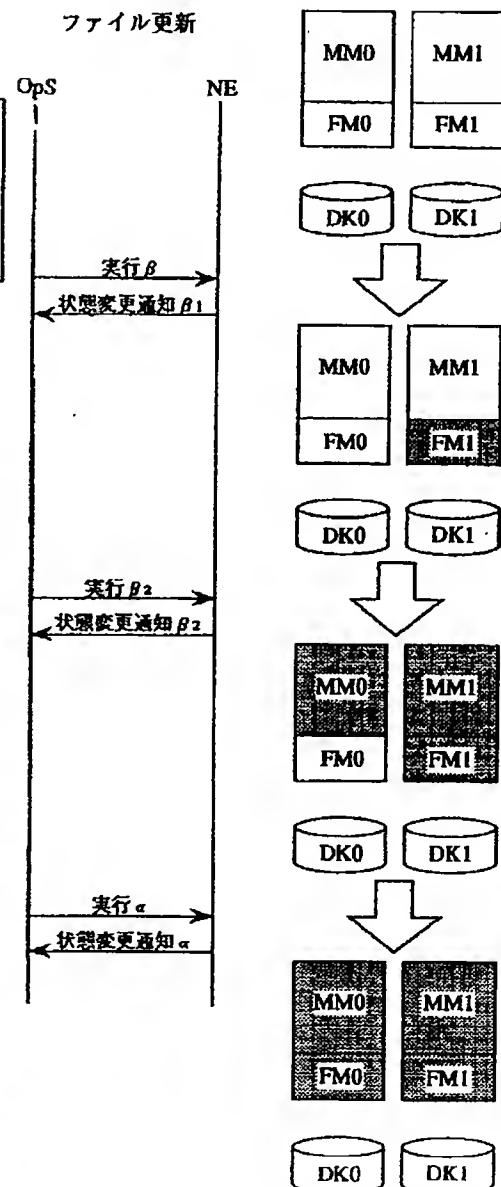
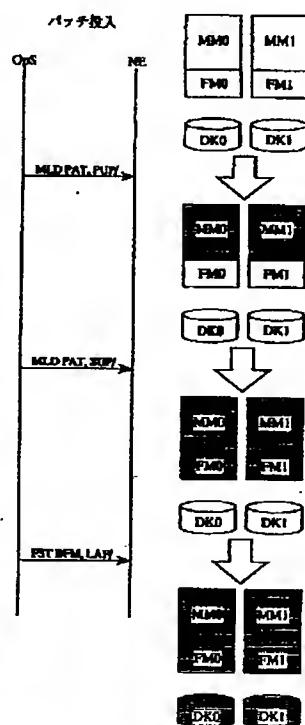
〔四九〕



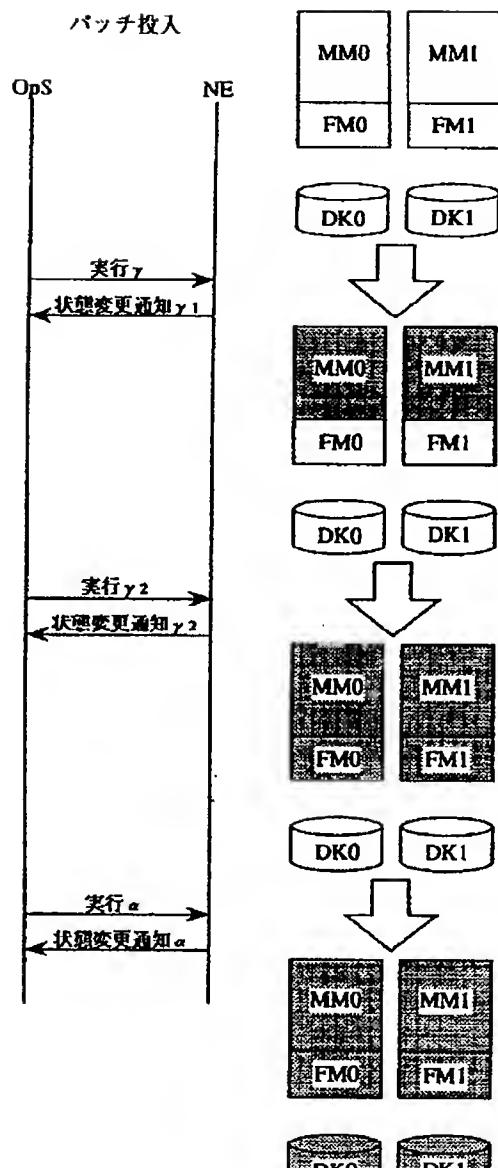
〔图15〕



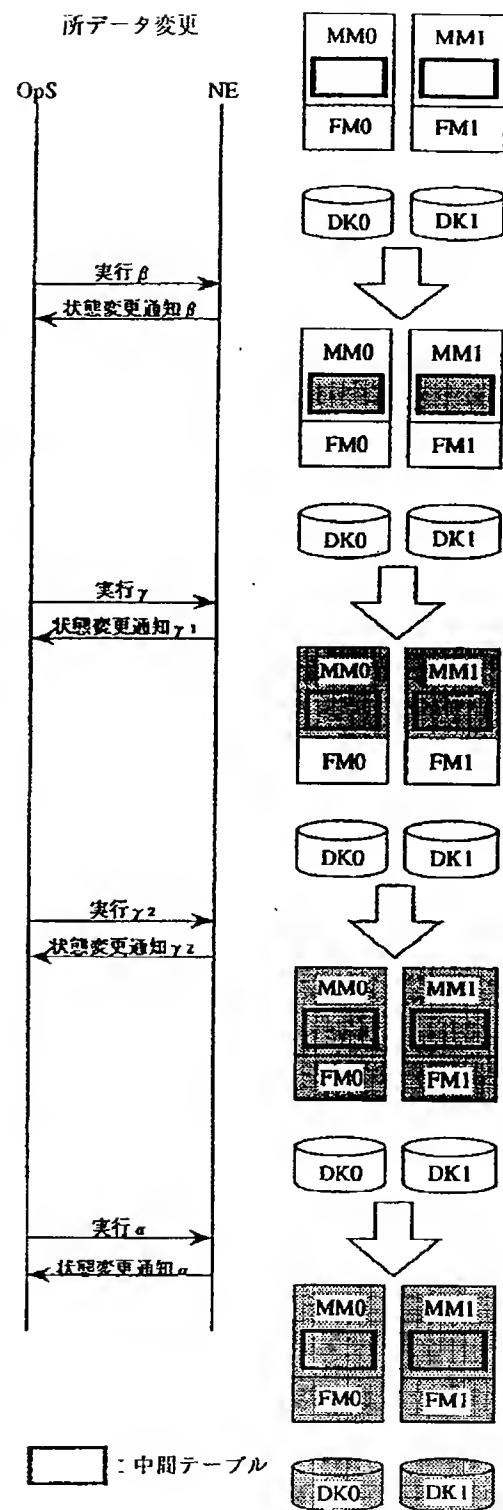
[图16]



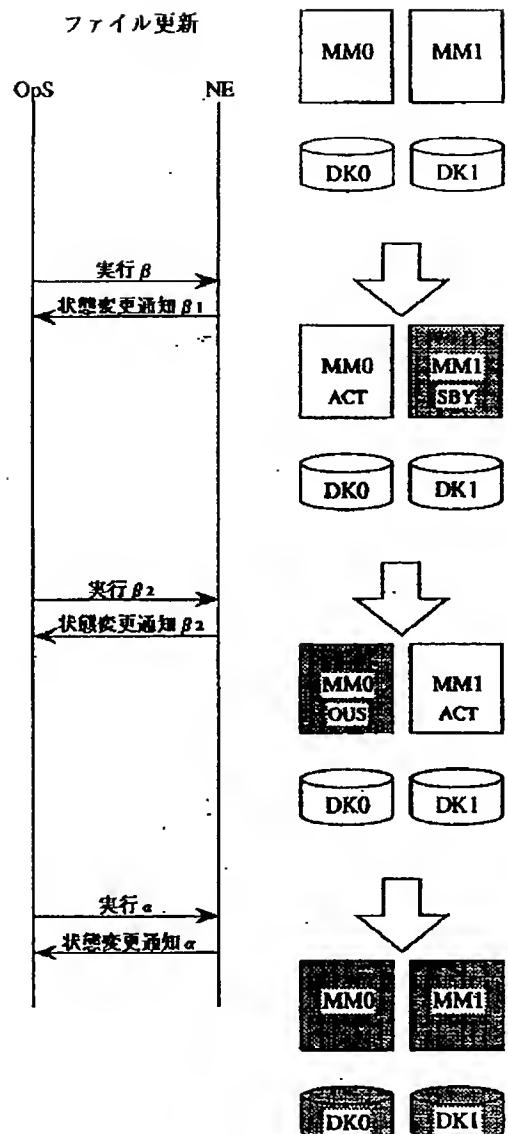
【図10】



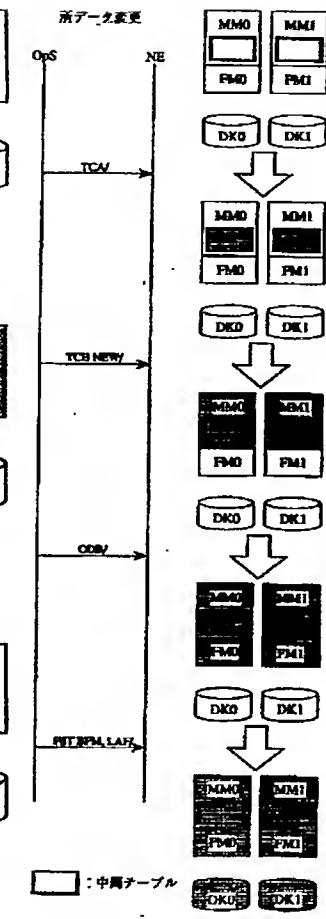
【図11】



【図12】



【図17】



【図18】

